

作业提交服务器: ftp://192.168.134.123

用户名: uploader

密码: sdt7%5252@3

! 注意事项

1. 每题需要提交一个cpp文件，确保cpp文件可以正确编译。cpp文件用题目编号命名。
2. 每题需要提交至少一张运行结果截图。将所有题目的截图放在一个pdf文件中。
3. 将所有cpp文件和一个pdf文件打包成zip文件，用自己的学号命名，并上传至ftp指定文件夹。

举例说明：比如学号是1001，本次有3个题目，那么最终提交1个1001.zip文件，其中包含3个cpp文件（分别是1.cpp, 2.cpp, 3.cpp）和1个pdf文件，其中pdf文件中包含至少3个运行结果的截图。第1次作业上传至hm1文件夹。

实践课第4次作业

提交时间：2024年6月3日22:00

1. 完善我们的链表设计，其中头文件list.hpp如下所示，请在源文件list.cpp中完成函数insert、add、erase、find和advance的定义。

```
1 //  list.hpp
2 #ifndef list_hpp
3 #define list_hpp
4
5 struct Link {
6     int value;
7     Link* prev;
8     Link* succ;
9     Link(int v, Link* p = nullptr, Link* s = nullptr)
10        : value { v }, prev { p }, succ { s } { }
11 };
12
13 Link* insert(Link* p, Link* n); //在p之前插入n, 返回n
14 Link* add(Link* p, Link* n); //在p之后插入n, 返回n
15 Link* erase(Link* p); //从链表中删除*p, 返回p的后继
16 Link* find(Link* p, int val); //在链表中查找val, 返回nullptr表示未找到
17 Link* advance(Link* p, int n); //n为正数表示前进, 负数表示后退
18
19#endif /* list_hpp */
```

然后基于如下的use.cpp文件使用链表。本题需提交list.cpp文件以及一个pdf文件，其中包含use.cpp运行结果截图。

```
1 // use.cpp
2
3 #include <iostream>
```

```

4 #include <vector>
5 #include "list.hpp"
6
7 //显示列表内容
8 void print(Link* p)
9 {
10     std::cout << "{";
11     while (p) {
12         std::cout << p->value;
13         p = p->succ;
14         if (p) std::cout << ", ";
15     }
16     std::cout << "}\n";
17 }
18
19 int main()
20 {
21     Link* a { new Link { 4 } };
22     a = insert(a, new Link { 3 });
23     a = insert(a, new Link { 2 });
24     a = insert(a, new Link { 1 });
25     std::cout << "The list is: ";
26     print(a);
27
28     std::vector<int> v {3, 5, 1, 4, 2};
29     for(auto e : v) {
30         Link* p = find(a, e);
31         if (p) {
32             if (p == a)
33                 a = p->succ;
34             erase(p);
35         }
36         std::cout << "after erase " << e << ":" ;
37         print(a);
38     }
39     return 0;
40 }
```

2. 基于我们设计的链表解决约瑟夫问题。

问题：假设有n个编号为1~n的小孩围成一圈，从编号为s的小孩开始计数，每次计数到k，则该小孩退出圆圈，然后跳过1个小孩，重新开始计数，如此继续直到剩下1个小孩，返回该小孩的编号。如果s大于n，则返回-1。请补充函数josephus，并编译运行程序。本题需要提交一个cpp文件和一个pdf文件，其中包含程序运行截图。

```

1 #include <iostream>
2 #include "list.hpp"
3
4 int josephus(int n, int s, int k)
5 {
6     // add your code here
```

```
7 }
8
9 int main()
10 {
11     // code to test the josephus function
12     for (int i { 5 }; i <= 10; ++i) {
13         for (int j { 1 }; j <= 7; ++j) {
14             for (int k { 1 }; k <= 3; ++k)
15                 std::cout << i << ' ' << j << ' ' << k << " : " << josephus(i, j,
16                 k) << std::endl;
17         }
18     }
19 }
```

3. 我们设计的链表实际上被称为双向链表。因为链表中每个节点都存放了指向其前驱和后继节点的指针，所以可以从两个方向访问链表。双向链表的缺点如下：第一，每个节点存放两个指针，空间开销大；第二，插入和删除节点更加费时，因为涉及到更多的指针操作。单向链表克服了这些缺点，每个节点仅仅存放指向其后继节点的指针。当然，此时我们只能从一个方向访问链表，即从前往后。考虑题目1中头文件list.hpp中声明的5个函数insert、add、erase、find和advance，针对单向链表给出可能的函数实现。如果某个函数不容易实现，你可以不实现，但请给出具体的原因，即为什么不容易实现。本题仅需提交一个pdf文件，其中包含对应的函数实现或者不容易实现的原因。